REMARKS/ARGUMENTS

Claims 1-38 are pending in this application. Claims 3-12 and 15 stand rejected under 35 U.S.C. § 102(b) as allegedly being anticipated by U.S. Patent No. 5,481,722 ("Skinner"). Claims 22-26 stand rejected under 35 U.S.C. § 102(b) as allegedly being anticipated by U.S. Patent No. 5,903,897 ("Carrier"). Claims 27, 29 and 31-38 stand rejected under 35 U.S.C. § 102(b) as allegedly being anticipated by U.S. Patent No. 5,649,200 ("Leblang"). Claims 1-2 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over U.S. Patent No. 6,223,343 ("Hopwood") in view of Skinner. Claims 16-21 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Skinner in view of Hopwood. Claims 13-14 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Skinner in view of U.S. Patent No. 5,481,722 ("Howard"). Claims 28 and 30 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Leblang. Applicant respectfully requests reconsideration of the present application in light of the above recited amendments and below recited remarks.

The Claimed Invention

The present invention discloses systems and methods for managing code changes for software development. More specifically it is disclosed that:

> "[a]n aspect of the invention provides better ergonomics and increased speed in merging versioned documents by indicating differences between them at multiple different subdivisions or levels, such as line and character levels.
>
> Another aspect of the invention increases the reliability of merges between documents by comparing them not only for incompatible changes with respect to each other, but also by detecting possible alternative histories from a common parent document, and flagging these as potential conflicts.
>
> Another aspect increases flexibility in reverting to previous versions by removing changes made during an earlier version to be backed out from a current version while retaining changes made in a version later than the removed version but earlier than the current version.
>
> A further aspect of the invention increases the ability to integrate all the material pertaining to a change, by keeping it together in one place. This aspect associates both versioned and nonversioned documents for the same version together, so that they can be manipulated as a single unit.
>
> Yet a further aspect permits developers to work in together in constructing a new build of the software even while another build is being

> tested and patched. Multiple copies of the documents are made in different areas. While one set of copies is built and tested, private copies for each builder have previous changes removed, so that they can modify clean copies of the documents (Application, Summary of the Invention)."

Rejections Under 35 U.S.C. § 102(b)

1. Claims 3-12 and 15 stand rejected under 35 U.S.C. § 102(b) as allegedly being anticipated by Skinner. Applicants respectfully traverse.

Skinner discloses a method and apparatus for merging change control delta structure files of a source module from a parent and a child development environment. Change deltas are created and propagated among the environments without any loss in change history (Skinner, Summary of the Invention).

With respect to independent claim 3, Skinner does not teach or suggest features recited in the claim, namely:

> "defining at least two nested types of subdivision in both of the documents; . . .
>
> for a current subdivision of the second type, comparing a current second-type subdivision of one of the documents to a current second-type subdivision of the other document and indicating any differences therebetween;
>
> repeating the comparing step for further second-type subdivisions of the documents within the current subdivision of the first type; and . . .
>
> producing an output document indicating the differences found in the comparing step.

Skinner, in fact, does not even mention two types of nested subdivisions. The Examiner analogizes the "start control line", "end control line", and "text line" of Skinner to the two types of nested subdivisions as recited in the claim. Applicants respectfully disagree with this analogy. A type of nested subdivision is a type of subdivision within a document. For example, a line may be a first type of nested subdivision and a character may be a second type of nested subdivision. ***The "start control line", "end control line", and "text line" are all lines and, therefore, are merely three instances of a single line type subdivision.*** As Skinner does not even mention a second type of nested subdivision, Skinner cannot possibly teach defining and comparing second type subdivisions and producing an output document indicating the differences in the comparisons.

Independent claim 8 recites similar limitations as independent claim 3 directed to three types of nested subdivisions. Applicants respectfully submit that independent claim 8 is therefore patentable for the same reasons.

With respect to independent claim 11, Skinner does not teach or suggest features recited in the claim, namely:

> comparing both child documents with the common parent document and indicating any possible conflicts between the child documents for portions of the child documents that are the same as each other; and
>
> producing a merged output document indicating both the actual and the possible conflicts.

In fact, Skinner teaches away from comparing children to a common parent. Skinner discloses that two children may store a first revision of a parent. When a first child is reconciled with the parent, the modifications made in the first child are compared with the first revision of the parent, and the modifications to the first revision are incorporated into a second revision of the parent. When a second child attempts to reconcile with the parent, the modifications made in the second child are resynchronized and compared with the second revision of the parent. Thus, Skinner discloses that the first child is compared to the first revision of the parent while the second child is compared to the second revision of the parent (Skinner, Fig. 8; Col. 12, lines 29-51). The first revision of the parent and the second revision of the parent are clearly not a "common parent".

Applicants respectfully submit that dependent claims 4-7, 9, 10, and 12 are patentable at least by reason of their dependency.

2. Claims 22-26 stand rejected under 35 U.S.C. § 102(b) as allegedly being anticipated by Carrier. Applicants respectfully traverse.

Carrier discloses a software documentation release control system. A source module is attached to a corresponding release form primarily by engineers or developers. The release form is then assigned to a build. Documentation about the build such as, for example, a build log and a build report are stored in a project file directory (Carrier, Summary of the Invention).

With respect to independent claim 22, Carrier does not teach or suggest features recited in the claim, namely: "retrieving both versioned and nonversioned documents as a single unit." The Examiner asserts that source files, check out information, and defect tracker of Figures 11 and 12 are analogous to this feature. Applicants respectfully submit that checking out and tracking defects in source files clearly does not teach or suggest retrieving both versioned and nonversioned documents as a single unit. Carrier fails to even mention a single unit including versioned and nonversioned documents.

Applicants respectfully submit that dependent claims 23-26 are patentable at least by reason of their dependency.

3. Claims 27, 29 and 31-38 stand rejected under 35 U.S.C. § 102(b) as allegedly being anticipated by Leblang. Applicants respectfully traverse.

Leblang discloses a dynamic rule-based version control system. Derived objects may be stored in a public VOB area or a private view corresponding to particular users. All objects in the private view are initially read only objects, but, if the users wish to edit the objects, then the objects may be "checked out". Upon completion of the editing, the objects may be "checked in". Each derived object may have a corresponding configuration record that contains an audit of the build process used to create the object (Leblang, Summary of the Invention).

The Examiner analogizes the configuration record of Leblang to the association file recited in independent claim 27. However, Leblang does not teach or suggest features of the association file recited in claim 27, namely:

> a plurality of entries each ***designating*** a version of one of the versioned documents;
> at least one entry ***designating*** a nonversioned document pertaining to at least one of the versioned documents.

Although the configuration record of Leblang ***audits*** elements previously used in a build, the configuration record does not ***designate*** elements as required by claim 27. Designating versions in the association file ***causes*** the versions to be incorporated in a change set. By contrast, auditing an element in the configuration record ***does not cause*** the element to be incorporated in a build. The configuration record is not generated until the build occurs and

is merely a record of elements ***that have already been incorporated*** into the build (Leblang, Col. 25, lines 30-42).

With respect to independent claim 31, Leblang does not teach or suggest features of the claim, namely:

> "adding a set of build specific changes to files in the enlistment area; and . . . thereafter, removing the build-specific changes from the enlistment files . . ."

The Examiner fails to state where "build specific changes" is mentioned with respect to the check in / check out procedure of Leblang. Leblang arguably discloses making local changes to files, but Leblang does not even mention adding and removing build specific changes.

Applicants respectfully submit that dependent claims 29 and 32-38 are patentable at least by reason of their dependency. Accordingly, reconsideration and withdrawal of the 35 U.S.C. § 102(b) rejections are respectfully requested.

Rejections Under 35 U.S.C. § 103(a)

4. Claims 1-2 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Hopwood in view of Skinner. Applicants respectfully traverse.

Hopwood discloses a computer system and method to track and control element changes throughout application development. A revision management system includes an inventory group manager (IGM), a work group manager (WGM), and a version manager (VM). The IGM generates and manages one or more lists for each of one or more elements logically related to each other as one or more inventory groups. The IGM also maintains and manages the lists of the elements in the inventory group and provides new elements to be added to the inventory group. The WGM selects the elements relating to a project that require modification. and selects elements from the inventory groups to form a project work group. The VM tracks and manages the modifications to the elements in the project work group, and generates reports with respect to the modifications responsive to user selectable criteria (Hopwood, Summary of the Invention).

Applicants respectfully submit that the combination of Hopwood and Skinner do not teach or suggest features recited in claim 1, namely:

> "comparing multiple ones of the versioned documents at a plurality of different subdivision levels and indicating the changes on an output document at each subdivision level;
>
> comparing the multiple ones of the versioned documents to a common parent document and indicating possible conflicts therebetween caused by alternative histories from the parent document;
>
> unmerging from a later version of one of the versioned documents a set of changes previous to a further set of changes; and . . .
>
> updating one or more files by copying them from a common storage area to a private enlistment area, adding build-specific changes to the enlistment copies, making local changes to the modified enlistment copies, and thereafter removing the build-specific changes and returning the files to the common area."

With respect to the first "comparing" step of claim 1, the Examiner analogizes the element hierarchy of Hopwood (Fig. 11) to the "subdivision levels" of a document as recited in claim 1. Applicants respectfully disagree with this analogy. Document subdivisions are subdivisions ***within a document*** such as, for example, a line of text or a character of text. Hopwood arguably discloses that ***entire elements*** may be compared as part of a project organization, but Hopwood does not even mention ***a plurality*** of levels of subdivisions ***within an element***.

Additionally, the Examiner analogizes the "unmerging" step of claim 1 to "regression, restore, and revert" as disclosed by Hopwood. "Regression" is defined by Hopwood as "overlaying of one developer's changes by another developer (Hopwood, Col 28, lines 45-46)." Thus, regression requires unmerging only those of the one developer's changes that are affected by the other developers changes. By contrast, unmerging a previous set of changes requires unmerging the entire previous set of changes even if some of the changes are not affected by future changes. Furthermore, "reverting" and "restoring" require replacing current versions with previous versions rather than removing previous versions (Hopwood, Col 28, lines 51-55).

Additionally, the Examiner analogizes the "updating" step of claim 1 to checking in and checking out files from a common area as disclosed by Hopwood. The check in / check out procedure of Hopwood does not include or suggest adding a set of build specific changes to files in an enlistment area; and removing the build-

specific changes from the enlistment files after making local changes to the enlistment files. The Examiner fails to state where "build specific changes" is mentioned with respect to the check in / check out procedure of Hopwood. The Examiner seems to imply that "build specific changes" and "local changes" are the same.

Additionally, with respect to the second "comparing" step of claim 1, Skinner does not teach or suggest comparing children to a ***common*** parent as discussed above with reference to claim 11.

Applicants respectfully submit that dependent claim 2 is patentable at least by reason of its dependency.

5. Claims 16-21 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Skinner in view of Hopwood. Applicants respectfully traverse.

Skinner in view of Hopwood do not teach or suggest the selective unmerging feature recited in independent claims 16 and 20. The Examiner admits that Skinner does not disclose this feature (Office Action, page 10, paragraph 5). Furthermore, as discussed above with reference to claim 1, "regression, restore, and revert" as disclosed by Hopwood is not analogous to this feature.

Applicants respectfully submit that dependent claims 17-19 and 21 are patentable at least by reason of their dependency.

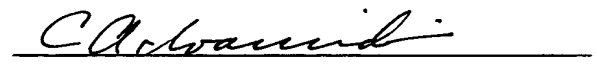
6. Claims 13-14 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Skinner in view of U.S. Patent No. 5,481,722 ("Howard"). Applicants respectfully traverse. Applicants respectfully submit that claims 13 and 14 are dependent from independent claim 11 and are therefore patentable for the same reasons.

7. Claims 28 and 30 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Leblang. Applicants respectfully traverse. Applicants respectfully submit that claims 28 and 30 are dependent from independent claim 27 and are therefore patentable for the same reasons. Accordingly, reconsideration and withdrawal of the 35 U.S.C. § 103(a) rejections are respectfully requested.

CONCLUSION

In view of the above remarks, Applicants respectfully submit that the present application is in condition for allowance. Reconsideration of the application and an early Notice of Allowance are respectfully requested.

Date: December 16, 2003



Christos A. Ioannidi
Registration No. 54,195

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439